Implementation and Experiences using *hypre* in TEMPEST and COGENT

BOUT++ 2013 Workshop

September 4, 2013

Milo Dorr

Lawrence Livermore National Laboratory

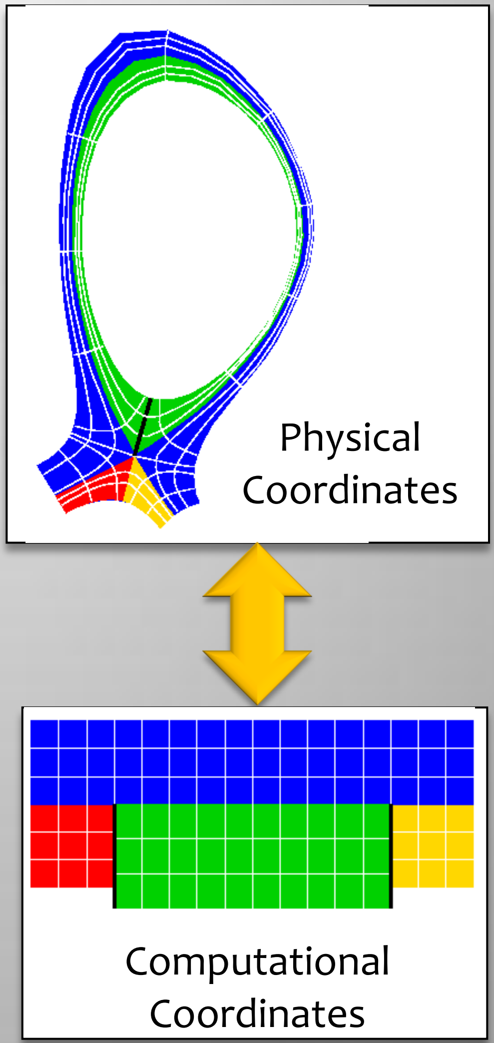
LLNL-PRES-643264

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Premise: BOUT++ real space solver needs may be similar to those of TEMPEST and COGENT

- Kinetic edge plasma simulation codes:
  - **TEMPEST**: Developed as part of an LDRD SI collaboration between the LLNL Fusion Energy Program (FEP) and Computation (CASC)
  - **COGENT**: Currently under development as part of the Edge Simulation Laboratory (ESL) collaboration between the DOE ASCR Applied Mathematics Research program and FES theory program

- Similarities:
  - Based on continuum gyrokinetic models
  - Use coordinate mapping to block structured, locally rectangular computational grids to accommodate strong anisotropy
  - Built on structured AMR libraries
    - TEMPEST is buildable on SAMRAI or Chombo
    - COGENT is written in native Chombo
  - Use ***hypre*** to solve various linear systems



Gyrokinetic systems couple Boltzmann and Maxwell equations in a 4D or 5D phase space

Gyrokinetic Boltzmann:

$$\frac{\partial}{\partial t}\left(B_\parallel^* f\right) + \nabla_{\dot{\mathbf{R}}}\left(\dot{\mathbf{R}} B_\parallel^* f\right) + \frac{\partial}{\partial_{v_\parallel}}\left(\dot{v}_\parallel B_\parallel^* f\right) = C(f,f)$$

describes the evolution of a distribution function

$$f \equiv f(\mathbf{R}, v_\parallel, \mu, t)$$

in gyrocenter phase space coordinates

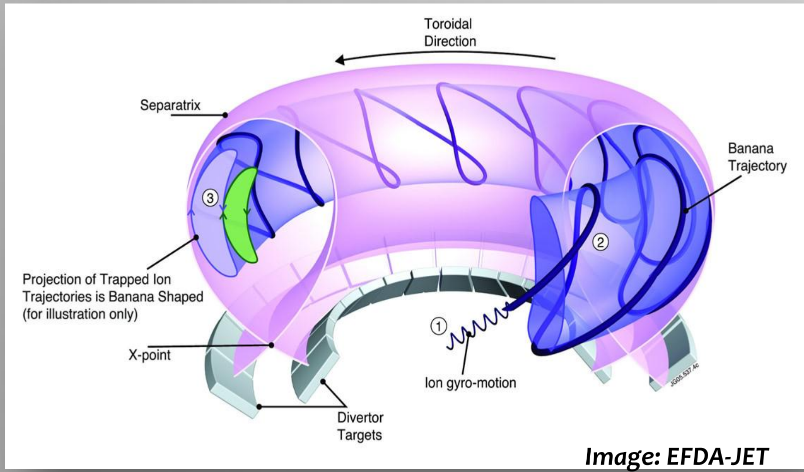


Image: EFDA-JET

Gyrokinetic Poisson (long wavelength limit):

$$-\nabla \cdot \left((De)^2 \mathbf{I} + \frac{(La)^2}{B^2} \sum_i Z_i m_i \bar{n}_i \left(\mathbf{I} - \mathbf{b}\mathbf{b}^{\mathbf{T}}\right) \right) \nabla \Phi = \sum_i Z_i \bar{n}_i - n_e$$

gyro-averaged ion density

gyrophase-dependent ion density

$\mathbf{B}$	Magnetic field
Φ	Potential
De	Debye number (normalized Debye length)
La	Larmor number (normalized gyroradius)

$$\dot{\mathbf{R}} \equiv \frac{v_\parallel}{B_\parallel^*}\mathbf{B}^* + \frac{La}{ZB_\parallel^*}\mathbf{b} \times \mathbf{G} \qquad \dot{v}_\parallel \equiv -\frac{1}{mB_\parallel^*}\mathbf{B}^* \cdot \mathbf{G}$$

$$\mathbf{B}^* \equiv \mathbf{B} + La\frac{mv_\parallel}{Z}\nabla_R \times \mathbf{b} \qquad B_\parallel^* = \mathbf{b} \cdot \mathbf{B}^*$$

$$\mathbf{G} \equiv Z\nabla_R \Phi + \frac{\mu}{2}\nabla_R|\mathbf{B}| \qquad \mathbf{b} \equiv \mathbf{B}/|\mathbf{B}|$$

TEMPEST performs an implicit integration of a differential algebraic system, requiring solvers for preconditioning

- GK Poisson and Boltzmann electron model treated as algebraic equations
- Implemented using the IDA module of Sundials
- Variable-order (up to 5), variable-step backward difference formulas based on local error estimates
- Newton-Krylov nonlinear solver
- GMRES solver for finite-difference Jacobian
- Block preconditioner:

Kinetic electrons:

$$P \equiv \begin{pmatrix} \frac{\alpha_0}{\Delta t} I & 0 & 0 \\ 0 & \frac{\alpha_0}{\Delta t} I & 0 \\ 0 & 0 & L_{GKP} \end{pmatrix} \begin{array}{c} f_i \\ f_e \\ \Phi \end{array}$$

Boltzmann electrons:

$$P \equiv \begin{pmatrix} \frac{\alpha_0}{\Delta t} I & 0 & 0 \\ 0 & I & H \\ 0 & -I & L_{GKP} \end{pmatrix} \begin{array}{c} f_i \\ n_e \\ \Phi \end{array}$$

α_0 = leading BDF coefficient

L_{GKP} = GK Poisson matrix

$$n_e = \frac{\langle n_i \rangle}{\langle \exp(\Phi/T_e) \rangle} \exp\left(\Phi/T_e\right)$$

$\langle \cdot \rangle$ = flux surface average

$H = \partial n_e / \partial \Phi$

hypre solvers used for the GKPoisson(-Boltzmann) block:

- PCG, GMRES, BiCGStab
- Split solver with structured multigrid solvers in each block (SMG, PFMG)
- Algebraic multigrid (BoomerAMG)

COGENT is being developed using and IMEX time integrator, requiring solvers for the implicit stage predictions

- Consider semi-discrete problem with stiff and non-stiff terms:

$$\frac{du_i}{dt} = F_E(u_i) + F_I(u_i)$$

- General additive partitioned ARK_2 scheme

$$\left(u^{(s)} - \Delta t \gamma F_I(u^{(s)})\right) = u^n + \Delta t \sum_{j=1}^{s-1} \left[a_{s,j}^{[E]} F_E(u^{(j)}) + a_{s,j}^{[I]} F_I(u^{(j)})\right]$$
$$u^{n+1} = u^{(s)} + \Delta t \sum_{j=1}^{s} (b_j - a_{s,j}^{[E]}) F_E(u^{(j)})$$

- ARK4(3)6L[2]SA of Kennedy and Carpenter
  - Combines 4th-order Explicit RK with 4th--order Explicit Singly Diagonally Implicit RK
  - ESDIRK advantages: L-stability, stiff accuracy, stage order of two
  - Suffers from order-reduction in transition between stiff and non-stiff limits
  - Chombo provides an interface with dense output for time refinement
- With this framework, can try other IMEX RK schemes as well

TEMPEST uses standard second-order centered differencing of the GK Poisson (or GK Poisson-Boltzmann) operator

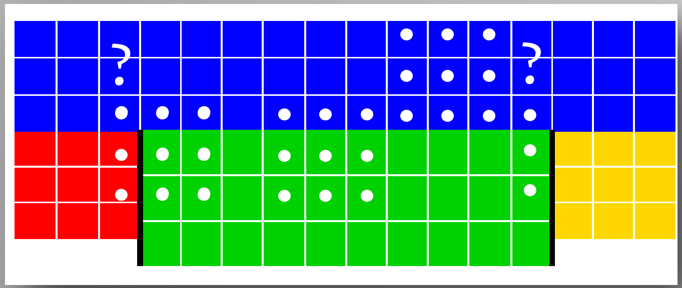
Nine-point stencil:

$a_{-1,1}$	$a_{0,1}$	$a_{1,1}$
$a_{-1,0}$	$a_{0,0}$	$a_{1,0}$
$a_{-1,-1}$	$a_{0,-1}$	$a_{1,-1}$

The *hypre* Struct interface is designed specifically for matrices resulting from stencil-based operators:

int offsets[][2] = {{-1,-1}, {0,-1}, {1,-1},
{-1,0}, {0,0}, {1,0},
{-1,1}, {0,1}, {1,1}};

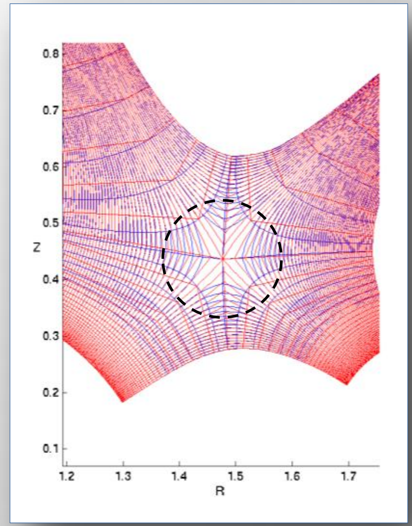
double values[] = {$a_{-1,-1}$, $a_{0,-1}$, $a_{1,-1}$,
$a_{-1,0}$, $a_{0,0}$, $a_{1,0}$,
$a_{-1,1}$, $a_{0,1}$, $a_{1,1}$};



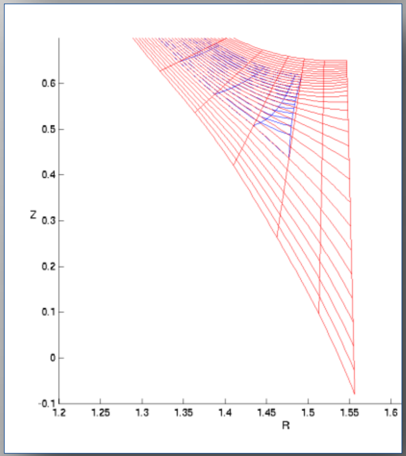
The *hypre* SStruct interface extends the specification to the union of structured blocks through the use of "neighbor blocks" that identify cells across block boundaries.

Differentiation in coordinates aligned with flux surfaces near the X point is problematic

- In axisymmetric edge geometry, flux surfaces (projected field lines) become progressively "kinked" approaching the X point
- Flux surface alignment is not needed near the X point, since the poloidal field component vanishes there
- Near the X point in each block, COGENT transitions to non-aligned mapping that is smooth up to and through the X point



Coordinate mapping modifications near the X point (before, after)



Extended left core block

COGENT employs a systematic formalism for high-order, mapped-grid finite volume discretizations

Cartesian coordinates:

Spatial domain discretized as a union of rectangular control volumes

$$V_{\mathbf{i}} = \prod_{d=1}^{D} \left[i_d - \frac{h}{2}, i_d + \frac{h}{2} \right]$$

Mapped coordinates:

Smooth mapping from abstract Cartesian coordinates into physical space

$$\mathbf{X} = \mathbf{X}(\boldsymbol{\xi}), \qquad \mathbf{X} : [0,1]^D \rightarrow \mathbb{R}^D$$

Fourth-order flux divergence average from fourth-order cell face averages:

$$\int_{\mathbf{X}(V_{\mathbf{i}})} \nabla_{\mathbf{X}} \cdot \mathbf{F} d\mathbf{x} = \sum_{\pm=+,-} \sum_{d=1}^{D} \pm \int_{A_d^{\pm}} \left(\mathbf{N}^T \mathbf{F} \right)_d d\mathbf{A}_{\boldsymbol{\xi}} = h^{D-1} \sum_{\pm=+,-} \sum_{d=1}^{D} \pm F^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} + O\left(h^4\right)$$

where

$$F^d_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} = \sum_{s=1}^{D} \langle N_d^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} \langle F^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} + \frac{h^2}{12} \sum_{s=1}^{D} \left(\mathbf{G}_0^{\perp,d} \left(\langle N_d^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} \right) \right) \cdot \left(\mathbf{G}_0^{\perp,d} \left(\langle F^s \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} \right) \right)$$

$\mathbf{G}_0^{\perp,d} =$ **second-order accurate centered difference of** $\nabla_{\boldsymbol{\xi}} - \mathbf{e}^d \dfrac{\partial}{\partial \xi_d}$

$$\langle q \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}^d} \equiv \frac{1}{h^{D-1}} \int_{A_d} q(\boldsymbol{\xi}) d\mathbf{A}_{\boldsymbol{\xi}} + O\left(h^4\right)$$

Free streaming is preserved:

$$\int_{A_d} N_d^s d\mathbf{A}_{\boldsymbol{\xi}} = \sum_{\pm=+,-} \sum_{d' \neq d} \pm \int_{E_d^{\pm},d'} M^s_{d,d'} d\mathbf{E}_{\boldsymbol{\xi}} \quad \Longrightarrow \quad \int_{\mathbf{X}(V_{\mathbf{i}})} \nabla_{\mathbf{X}} \cdot \mathbf{F} d\mathbf{x} = 0 \text{ for } \mathbf{F} \text{ constant}$$

[Colella, P. et al. (2011) *J. Comput. Phys.* **230** 2952-2976]

COGENT's mapped multiblock finite-volume discretization utilizes high-order interpolation at interblock boundaries

- To find the cell average of ϕ in a neighbor block ghost cell (centered at the red dot), assume a polynomial around the center:

$$\phi(\xi) = \sum_p a_p \xi^p$$

- Solve least squares system for coefficients
- Average interpolant over red cell

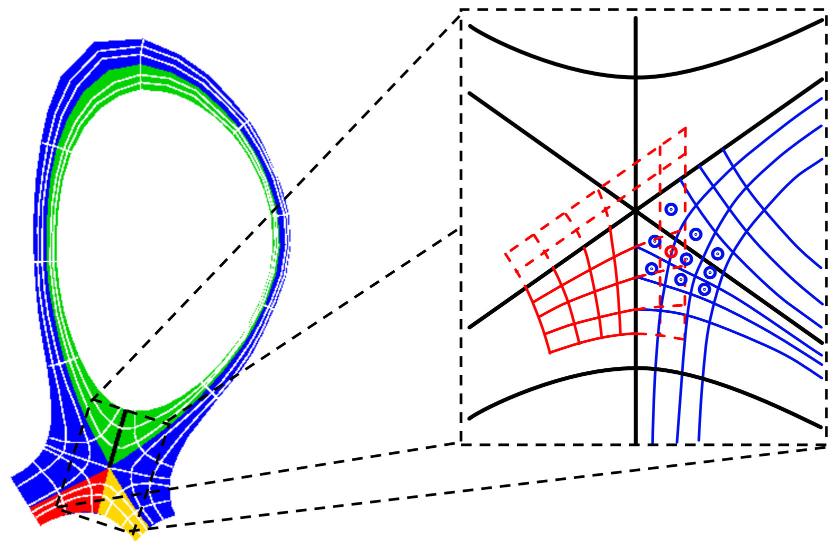
known for control volumes centered at blue dots

computable from (*i.e.*, requires) red inverse mapping

$$\int_V \phi(\xi)d\xi = \sum_p a_p \int_V \xi^p(\xi)d\xi$$

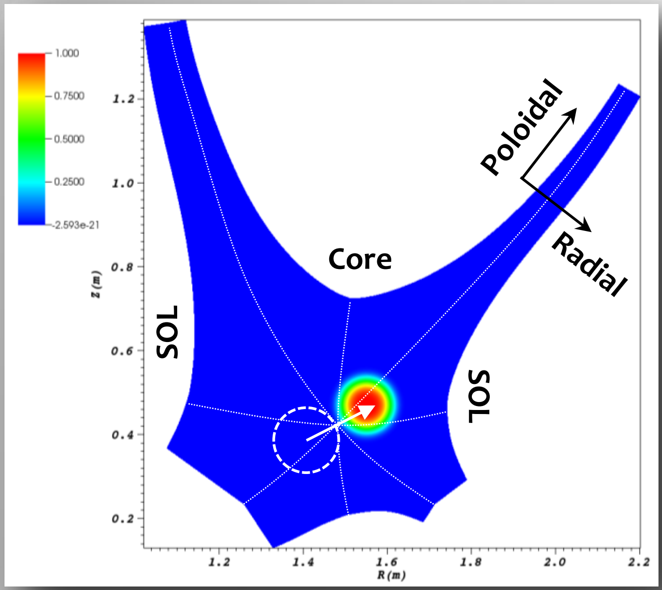
[McCorquodale, P. Chombo Mapped Multiblock Design Document]

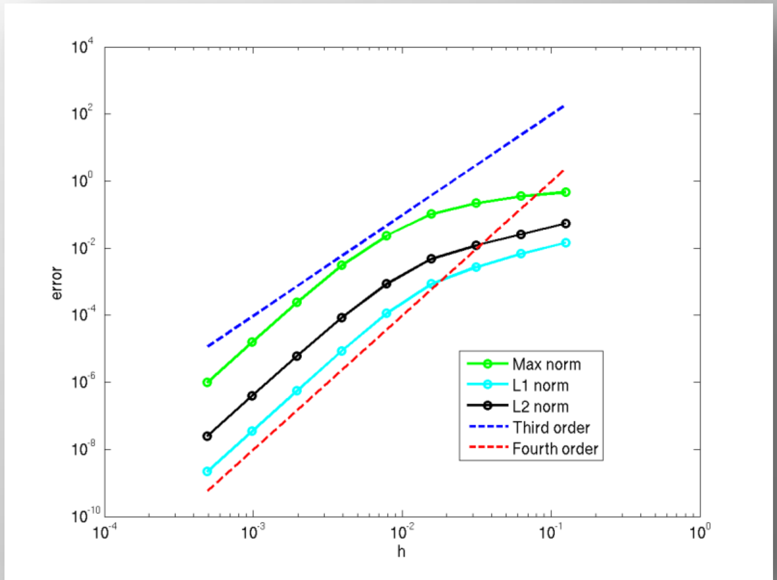
Interblock ghost cell interpolation



- Averaging of exchanged fluxes ensures strict conservation
- Interblock interpolation is performed entirely within Chombo mapped multiblock objects

COGENT maintains fourth-order accuracy for advection through the X-point

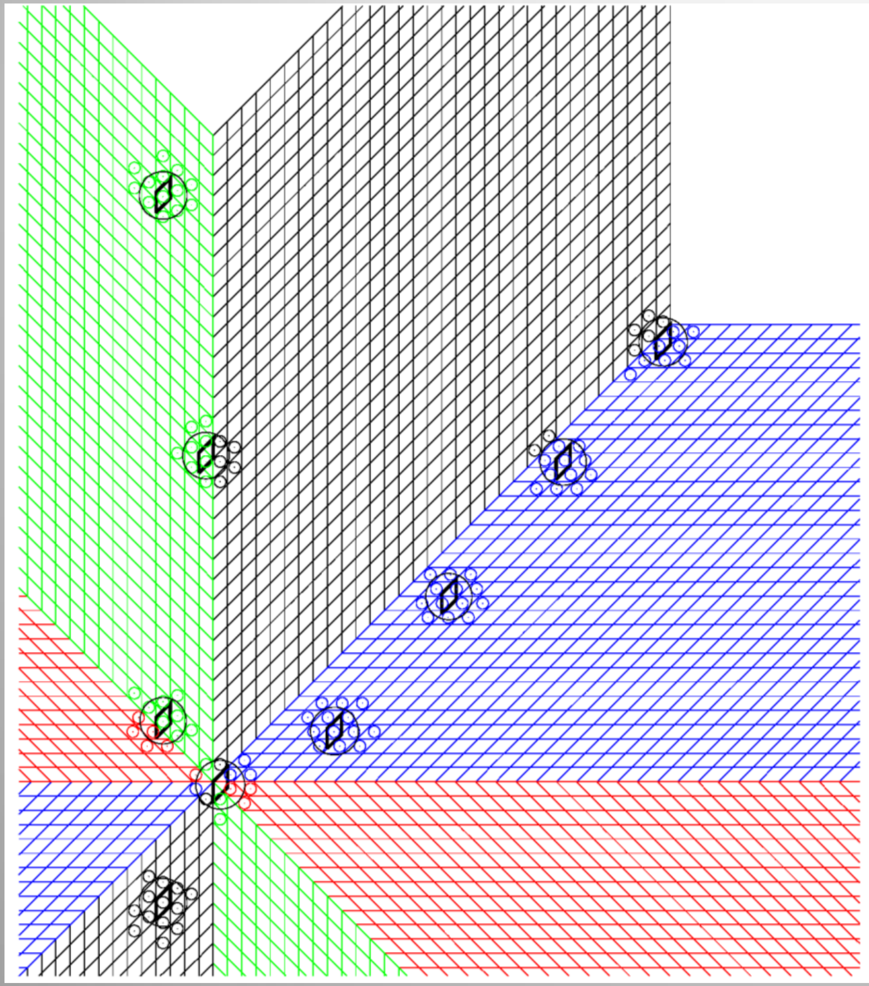




- Constant linear advection in configuration space through the X-point in 2D
- Non-axisymmetric in order to compare with analytic solution
- Grid convergence study across 9 resolutions by factor of 2

Domain	Resolution Sequence
Poloidal (in each block)	4, 8,..., 1024
Core Radial	8, 16,..., 2048
SOL Radial	12, 24,..., 3072

We have developed an interface between Chombo mapped multiblock geometries and *hypre* semi-structured solvers



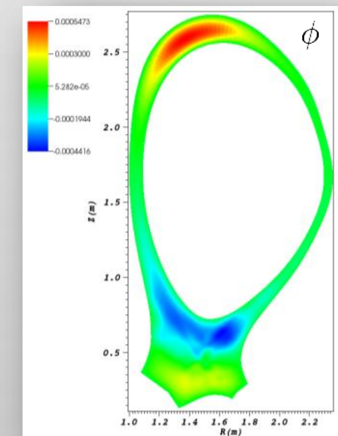
- The hypre SStruct interface constructs matrices in two steps
  - Structured stencil: Regular couplings within blocks, e.g. a nine-point stencil
  - Unstructured stencil: Sparse couplings at interblock boundaries
- We have created an interface to generate SStruct matrices (for tensor diffusion operators) and vectors from Chombo mapped multiblock geometries
  - *MultiblockLevelExchange* objects provide the stencil indices and weights needed to construct the sparse interblock couplings
  - *BlockRegister* objects provide a mechanism for exchanging stencil information at interblock boundaries needed to construct the conservative operators
  - As part of the SciDAC FASTMath institute, we are generalizing this interface to also interoperate with PETSc solvers

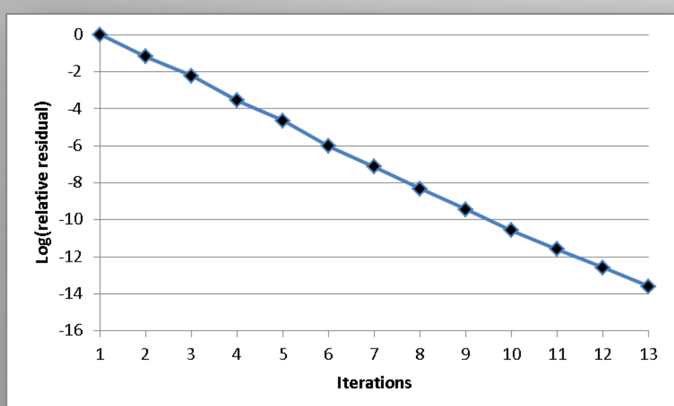
hypre BoomerAMG-preconditioned Krylov iteration is working well in COGENT

Divergence cleaning solve:

$$\Delta\phi = \nabla \cdot \mathbf{B}$$

- PCG
- Preconditioner: 2 BoomerAMG V-cycles with 2nd order operator

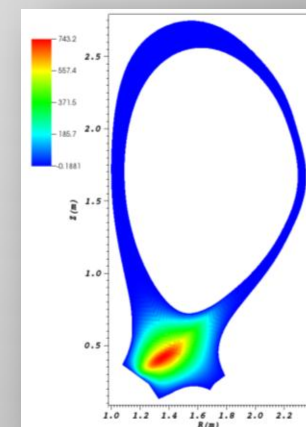


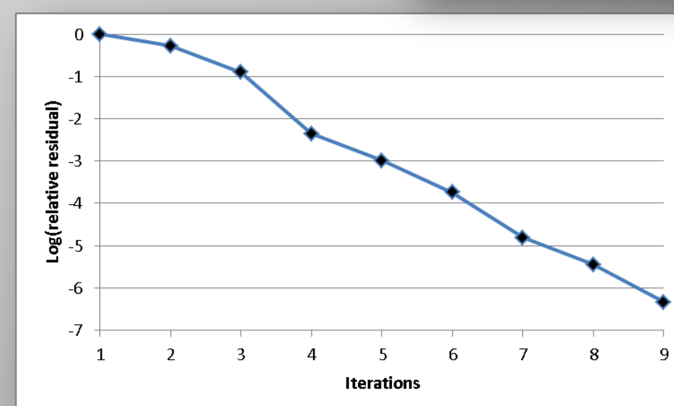


GKPoisson solve:

$$\nabla \cdot \left(\left[\lambda_D^2 \mathbf{I} + \lambda_L^2 \sum_i \frac{Z_i \bar{n}_i}{m_i \Omega_i^2} (\mathbf{I} - \mathbf{b}\mathbf{b}^T) \right] \nabla \Phi \right) = n_e - \sum_i Z_i \bar{n}_i$$

- PCG
- Preconditioner: 1 BoomerAMG V-cycle with 2nd order operator





Lawrence Livermore National Laboratory